**GAME 8 PUZZLE NGƯỜI CHƠI VỚI MÁY**

**Lâm Hoàng An 1 , Đặng Nguyễn Hoài Thư 1**

1 *Trường đại học Sư Phạm Kỹ Thuật thành phố Hồ Chí Minh*

**TÓM TẮT:**

Game 8 puzzle là một trò chơi gồm một bảng có n -1 ô vuông, tức là sẽ có một ô trống trong bảng. Các ô liền kề có thể di chuyển đến ô trống này. Từ một trạng thái bất kì, người chơi cần di chuyển các ô của bảng về một trạng thái đích nào đó với số lần di chuyển càng ít càng tốt. Từ những kiến thức đã học, nhóm em đã viết một chương trình game 8 puzzle người chơi với máy tính có thể thực thi được, giúp máy tính đánh tốt hơn với thời gian nhanh hơn. Chương trình được viết theo phương pháp lập trình hướng đối tượng, ngôn ngữ là C++. Chương trình sử dụng thuật toán minimax-cắt tỉa nhánh kết hợp với độ sâu, A\* và thuật toán expectiminimax. Chương trình được thiết kế giao diện dễ nhìn, dễ sử dụng. Chương trình từ trạng thái đích , random các trạng thái ban đầu, mà hầu hết các trạng thái đều thực thi được. Máy tính trở nên thông minh hơn, có thể tối ưu hóa được nước đi của mình. Game mang lại tính giải trí cao, đồng thời giúp người chơi rèn luyện trí óc của mình.

Từ khóa: Game 8 puzzle; trạng thái đích; A\* ; Alpha – beta prunning; expectiminimax;

1. **Đặt vấn đề:**

Ngày nay, nhu cầu vui chơi giải trí ngày càng tăng cao, có rất nhiều trò chơi, ứng dụng giải trí xuất hiện ngày càng nhiều. Trong đó, game N-puzzle có lẽ là một trong những trò chơi quen thuộc với nhiều người trên thế giới. Nó được biết đến với nhiều phiên bản cũng như tên gọi khác nhau, ví dụ: 8 – puzzle, 15 – puzzle, Boss – puzzle, Mystic Square,… Nhìn chung nó đều thuộc loại game xếp hình hay đầy hộp,.. Bên cạnh đó, đây cũng là một game đơn giản đối với những người mới bắt đầu tiếp cận với môn Trí tuệ nhân tạo. Trong những năm gần đây, trí tuệ nhân tạo đã được ứng dụng nhiều trong thực tế cuộc sống, mà đặc biệt là trong những game giải trí, giúp cho máy tính có thể trở nên thông minh hơn, có thể chơi thắng được con người, giúp người chơi thử thách bản thân mình nhiều hơn. Ngoài chơi game để giải trí, người chơi còn có thể rèn luyện trí óc của mình với những màn đấu trí hấp dẫn giữa con người với máy tính. Từ những cơ sở đó, nhóm em đã viết chương trình AI cho máy tính chơi game 8 puzzle với người. Mỗi lần chơi sẽ tung đồng xu để chọn nước đi, nếu mặt VND thì đến lượt của máy tính được đi, mặt 500 thì đến lượt người chơi được đi. Các phần còn lại của bài báo cáo được tổ chức như sau. Phần 2 trình bày phương pháp được sử dụng trong chương trình, bao gồm trạng thái ban đầu, trạng thái đích, các thuật toán giải quyết bài toán. Phần 3 là trình bày kết quả thực thi được của chương trình, đưa ra các so sánh trên những trạng thái kiểm tra. Phần 4 đưa ra những giải pháp cải thiện cho chương trình, cũng như những phương pháp để nâng cấp, mở rộng cho trò chơi. Phần 5 là tổng kết, đánh giá lại chương trình này. Phần 6 là tài liệu tham khảo.

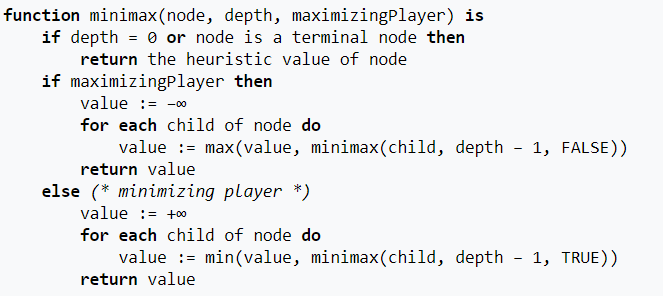
1. **Phương pháp:**
   1. **Trạng thái ban đầu, trạng thái đích:**

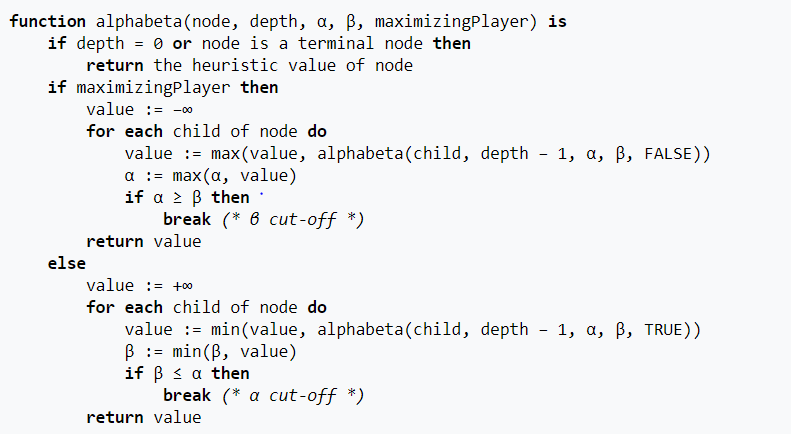
Trạng thái đích do người chơi tự thiết lập, từ trạng thái đích đã được nhập, chương trình sẽ di chuyển ô trống ngẫu nhiên khoảng 2000 lần để thiết lập trạng thái ban đầu.

* 1. **Các thuật toán sử dụng:**

Sử dụng thuật toán expectiminimax để random ngẫu nhiên mặt của đồng xu với xác suất là 0.5. Tính xác suất trong 2n số (chương trình áp dụng n=2), khi ở trạng thái xấu nhất, tăng xác suất xuất hiện mặt VND cho máy tính là 0.8, mặt 500 cho người chơi là 0.2, khi ở trạng thái gần với trạng thái đích, xác suất được thiết lập lại với máy tính là 0.5 và người chơi là 0.5.

Sử dụng thuật toán minimax để tối ưu hóa nước đi của máy tính, chương trình lấy max để lựa chọn nước đi tốt nhất cho máy tính, lấy min để lựa chọn nước đi gây khó khăn cho người chơi. Bên cạnh đó, chương trình áp dụng thuật toán alpha-beta prunning để loại bỏ đi các trường hợp xấu. Mã giả của thuật toán minimax và thuật toán alpha – beta prunning:

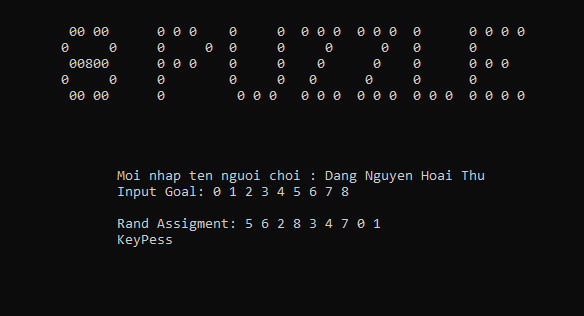


**Hình 2.** Mã giả của thuật toán minimax

**Hình 3.** Mã giả của thuật toán alpha – beta prunning

Sử dụng thuật toán A star để hạn chế thời gian chạy của thuật toán alpha – beta prunning, bằng cách cho thuật toán A star giải những trạng thái xấu, khi đến trạng thái tốt chương trình sẽ chạy alpha – beta prunning để tối ưu hóa nước đi của mình. Thuật toán sử dụng hàm h2 (lấy tổng giá trị hiện tại nhân với khoảng cách đi đến vị trị đúng) để lựa chọn trạng thái.

1. **Kết quả:**

Chương trình hoàn thành được yêu cầu đưa ra, chương trình chạy không lỗi, 100% thực thi được

**Hình 5.** Thiết lập trạng thái trò chơi

**Hình 6.** Giao diện chương trình đang thực thi

So sánh các thuật toán trong chương trình:

Trạng thái đích là như nhau cho 3 thuật toán: 0 1 2 3 4 5 6 7 8

|  |  |  |  |
| --- | --- | --- | --- |
| Trạng thái random ban đầu | Thuật toán A\* | Thuật toán Alpha – beta prunning | Thuật toán A\* kết hợp với thuật toán Alpha – beta prunning |
| 1 8 4 5 6 0 3 7 2 | 0.121 s | 1.016 s | 0.121 s |
| 5 3 6 0 2 7 1 4 8 | 0.164 s | 1.012 s | 0.164 s |
| 3 0 1 2 8 5 4 7 6 | 0.088 s | 1.002 s | 0.088 s |
| 1 4 3 0 2 6 7 8 5 | 0.143 s | 1.429 s | 0.143 s |
| 4 8 3 5 2 1 6 0 7 | 0.001 s | 1.835 s | 0.001 s |
| 7 2 8 4 5 6 1 0 3 | 0.096 s | 1.699 s | 0.096 s |

**Bảng 1:** So sánh thời gian chạy của 3 thuật toán

Thuật toán A\* có thời gian chạy nhanh hơn thuật toán Alpha – beta prunning, nhưng chỉ tìm được đường đi tốt nhất, nhưng không tìm ra được nước đi gây khó khăn cho người chơi.

Do đó, nhóm em đã kết hợp cả hai thuật toán A\* và thuật toán Alpha – beta prunning để máy tính có thể lựa chọn nước đi vừa tấn công, vừa phòng thủ. Khi ở trạng thái xấu, chương trình sẽ ưu tiên chạy thuật toán A\*, máy tính sẽ tìm ra trạng thái gần với đích hơn. Khi ở trạng thái gần với trạng thái đích, chương trình sẽ ưu tiên chạy thuật toán alpha-beta prunning .

Thời gian chạy thuật toán được tính trên một trong hai thuật toán A\* hoặc Alpha – Beta prunning nếu đang ở trạng thái xấu thì thời gian thực hiện sẽ bằng thời gian chạy thuật toán A\*. Còn nếu đang ở trạng thái gần trạng thái đích thời gian thực hiện sẽ bằng thời gian chạy thuật toán Alpha – Beta prunning.

1. **Thảo luận:**

Nhóm em đề xuất là cải tiến trò chơi có thể lưu được trạng thái trò chơi hiện tại của người chơi, để khi người chơi có việc đột xuất thoát game, người chơi có thể lựa chọn việc tiếp tục chơi cho lần đăng nhập tiếp theo. Bên cạnh đó, có thể cải tiến lại thuật toán để tối ưu hóa tốc độ xử lí và mức độ tiêu tốn tài nguyên của hệ thống. Nâng cấp chương trình thành NxN trạng thái thay vì chỉ 3x3 như hiện tại.

1. **Kết luận:**

Trong đồ án này, nhóm em đã áp dụng các thuật toán được học trong môn Trí tuệ nhân tạo để viết chương trình game 8 puzzle người chơi với máy. Chương trình được thiết kế giao diện dễ nhìn, dễ tiếp cận. Chương trình viết bằng phương pháp lập trình hướng đối tượng, nên việc sửa chữa, nâng cấp chương trình dễ dàng hơn. Việc kết hợp 2 thuật toán minimax alpha-beta prunning cùng với thuật toán A\* giúp tăng tốc độ xử lý của máy. Tuy nhiên, chương tình vẫn còn nhiều hạn chế. Thuật toán tỏ ra kém hiệu quả khi xử lý các trạng thái khó. Chương trình tiêu tốn nhiều tài nguyên hệ thống trong quá trình chạy. Cải tiến trò chơi có thể lưu được trạng thái trò chơi hiện tại của người chơi là hướng nghiên cứu tiếp theo của nhóm em trong đồ án này.

1. **Tài liệu tham khảo:**

[1]: Trần Đan Thư, Đinh Bá Tiến, Nguyễn Tấn Trần Minh Khang, *Lập trình hướng đối tượng*, nhà xuất bản Khoa học và kỹ thuật

[2]: Video vẽ bàn cờ trên console – Cong Tu Tran

<https://www.youtube.com/watch?v=AOvXmLpucXk> (10/3/2017)